Finding Stable Orientations of Assemblies with Linear Programming

David Baraff Raju Mattikalli Bruno Repetto
Pradeep Khosla

CMU-RI-TR-93-13

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

June 1993

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
June 1993

3. REPORT TYPE AND DATES COVERED
technical

4. TITLE AND SUBTITLE
Finding Stable Orientations of Assemblies with Linear Programming

5. FUNDING NUMBERS

6. AUTHOR(S)
David Baraff, Raju Mattikalli, Bruno Repetto and Pradeep Khosla

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER
CMU-RI-TR-93-13

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release;
Distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In the paper by Mattikalli et al.[5], the stability of an assemblage of frictionless contacting bodies with uniform gravity was considered. The problem of finding a stable orientation for such an assembly was formulated as a constrained maximin problem. A solution to the maximin problem yielded an orientation of the assembly that was stable under gravity; however, if no such orientation existed, then the solution to the maximin problem yielded the most stable orientation possible for the assembly. The maximin problem was solved using a numerical iteration procedure that solved a linear program for each step of the iteration. In this paper, we show that the stability problem can be considered a variant of standard zero-sum matrix games. A solution to the maximin problem can be found by solving a single linear program.

14. SUBJECT TERMS

15. NUMBER OF PAGES
16 pp

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unlimited	unlimited	unlimited	unlimited

Contents

List of Figures

Abstract

In the paper by Mattikalli *et al.*[5], the stability of an assemblage of frictionless contacting bodies with uniform gravity was considered. The problem of finding a stable orientation for such an assembly was formulated as a constrained maximin problem. A solution to the maximin problem yielded an orientation of the assembly that was stable under gravity; however, if no such orientation existed, then the solution to the maximin problem yielded the most stable orientation possible for the assembly. The maximin problem was solved using a numerical iteration procedure that solved a linear program for each step of the iteration. In this paper, we show that the stability problem can be considered a variant of standard zero-sum matrix games. A solution to the maximin problem can be found by solving a single linear program.

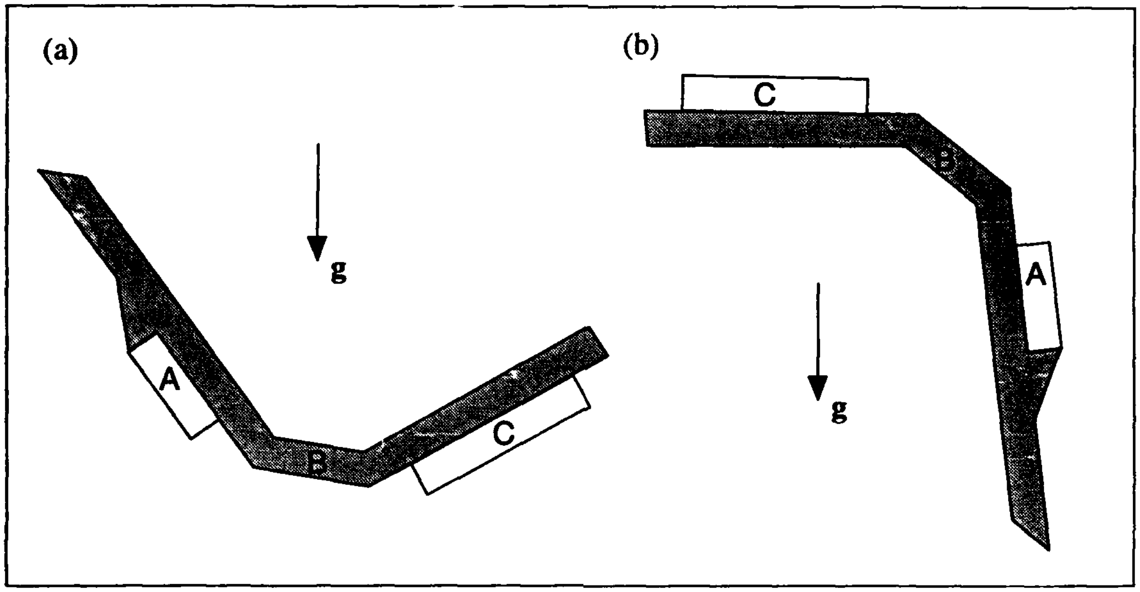

Figure 1: Body B is fixed in place, while bodies A and C are free to move. (a) The assembly is unstable for the current choice of $\mathbf{g}$. (b) The assembly has been reoriented so that it is stable.

1 Introduction

In Mattikalli *et al.*[5] we showed how a stable orientation for an assembly could be formulated as the solution to a constrained maximin problem. We define an assembly as a collection of frictionless contacting rigid objects, one or more of which is assumed to be fixed in place (for example a floor, a supporting surface such as a table, or an object held by a gripper). All objects in the assembly are initially motionless and are acted upon by an external force $m\mathbf{g}$ where m is an object's mass and $\mathbf{g}$ is a unit vector pointing straight down. If the objects remain motionless under the influence of the gravity field, we say the assembly is stable. Otherwise, the assembly is unstable.

An assembly that is unstable in one orientation might be made stable by a change of orientation (figure 1). Given an unstable assembly, we would like to be able to reorient the assembly to make it stable, if possible. However, rather than actually rotate objects in our frame of reference, we will instead choose a different unit gravity vector for which the assembly is stable, if such a gravity vector exists (figure 2). When we say we are searching for a stable orientation then, we mean that we are searching for a gravity direction $\mathbf{g}$ for which the assembly is stable. If it turns out that the assembly is unstable no matter what direction of gravity we choose, then we would like to determine an orientation (that is, a gravity direction) which makes the orientation as stable as possible. We will define the metric used to measure stability in section 5.

The formulation we use for finding the most stable orientation is based on energy considerations. If an assembly is initially at rest, the kinetic energy T of the assembly is zero. If the assembly is unstable and begins to move, then T must increase. Since the only

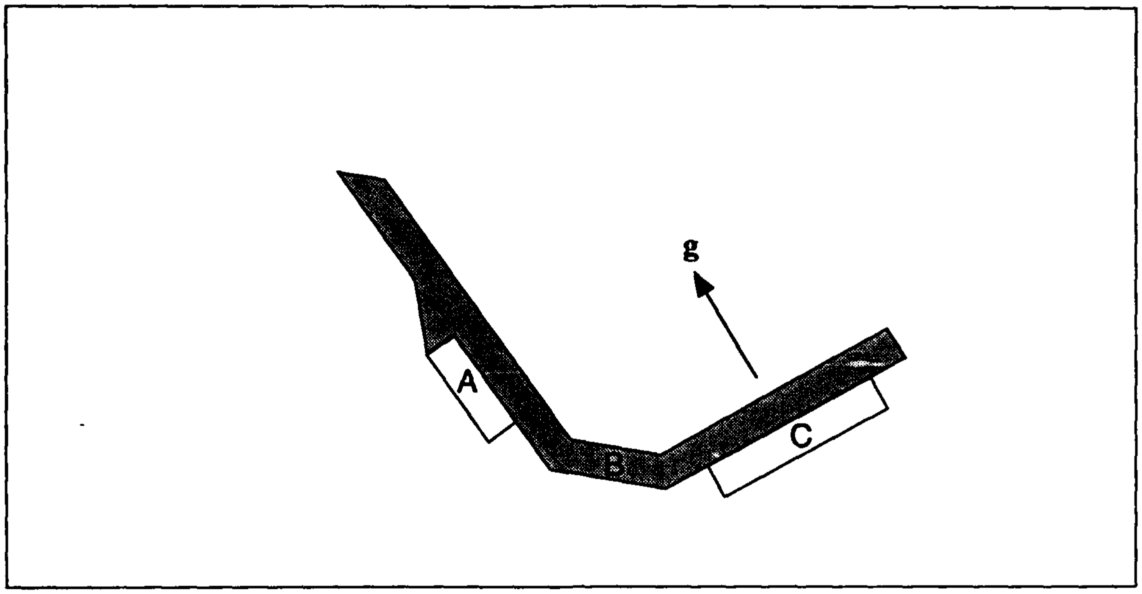

Figure 2: Rather than actually rotate objects, a new gravity direction **g** is chosen so that the assembly in figure 1a is stable.

external force is gravity, which is conservative, if T increases the potential energy U of the assembly must decrease. Thus, if every motion that does not cause interpenetration between objects is "uphill" (that is, causes U to *increase*), then the configuration is stable. Our goal then is to find a direction of gravity so that all allowable motions are "uphill." If this is not possible, we want to orient gravity to minimize the steepness of the most "downhill" allowable motion.

The optimal direction for **g** in Mattikalli *et al.*[5] was defined in terms of the solution to a constrained maximin problem. The maximin problem was solved by numerical iteration. Each step of the iteration involved solving a linear program. We will show that a solution to this maximin problem can be found by solving a single linear program, eliminating the need for an iterative solution method. The insight into this reduction lies in viewing the maximin problem as a variant of two-person zero-sum matrix games. Two-person zero-sum matrix games were first shown to be equivalent to linear programs by Dantzig. Mattikalli *et al.* discusses previous work on stability problems.

2 Motion Constraints

We will represent possible motions of an assembly in terms of virtual displacements. Let $\delta\mathbf{p}_i = (\delta\mathbf{r}_i, \delta\boldsymbol{\theta}_i)$ represent a displacement of the ith body in the system, with $\delta\mathbf{r}_i$ and $\delta\boldsymbol{\theta}_i$ vectors in $\mathbf{R}^3$. The vector $\delta\mathbf{r}_i$ denotes a translational displacement of the ith part, while $\delta\boldsymbol{\theta}_i$ denotes a rotation of magnitude $||\delta\boldsymbol{\theta}_i||$ of the body around its center of mass. The axis of the rotation is along the $\delta\boldsymbol{\theta}_i$ direction.

Because of contact between bodies, not all displacements are allowable. Consider

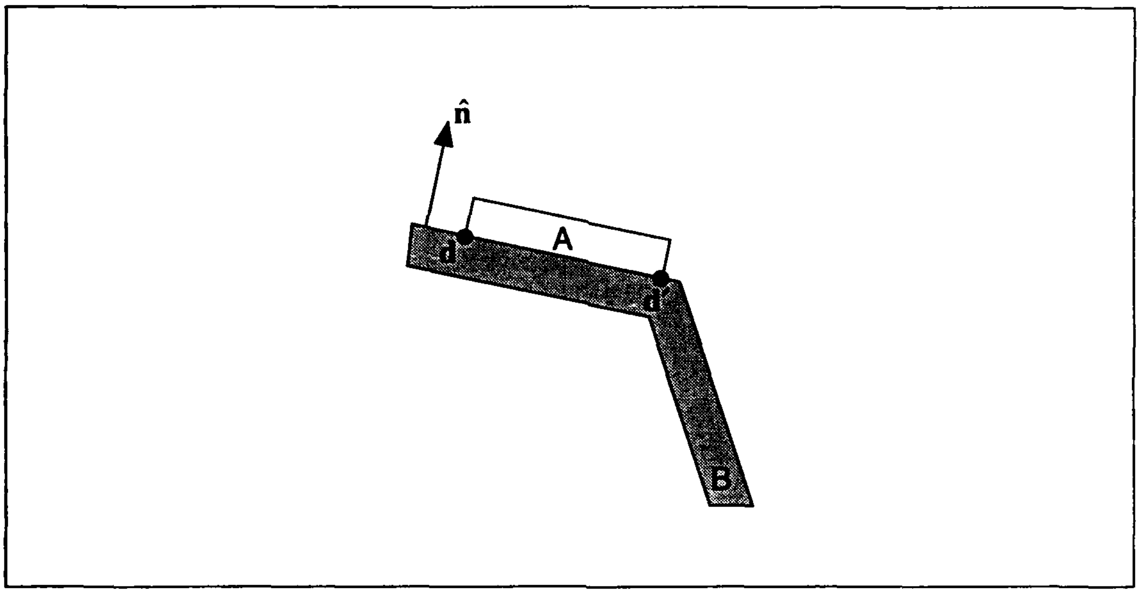

Figure 3: Contact between bodies A and B. Motion constraints are formulated in terms of the relative motion of the bodies at points $\mathbf{d}$ and $\mathbf{d}'$.

figure 3 where bodies A and B contact. If body A undergoes a displacement $\delta\mathbf{p}_a = (\delta\mathbf{r}_a, \delta\boldsymbol{\theta}_a)$, then point $\mathbf{d}$, as attached to body A, undergoes a particular displacement $\delta\mathbf{d}_a$. Similarly, a displacement $\delta\mathbf{p}_b$ of body B causes a displacement $\delta\mathbf{d}_b$ of point $\mathbf{d}$, as attached to body B. To prevent interpenetration from occurring, the relative displacement $\delta\mathbf{d}_a - \delta\mathbf{d}_b$ cannot have any component opposite the unit normal direction $\hat{\mathbf{n}}$. We can express this as the constraint

$$\hat{\mathbf{n}} \cdot (\delta\mathbf{d}_a - \delta\mathbf{d}_b) \geq 0. \tag{1}$$

Similarly, we also need to prevent interpenetration from occurring at point $\mathbf{d}'$ by requiring that $\hat{\mathbf{n}} \cdot (\delta\mathbf{d}'_a - \delta\mathbf{d}'_b) \geq 0$.

Note that we do not have to generate a constraint for *every* point of contact between two bodies. It is only necessary to place constraints on the vertices of the convex hull of the contact region. In this paper, we assume that all contact regions are polygons (or degenerate polygons), which means that we have to formulate constraints for only finitely many contact points. Baraff[2] discusses the difficulties in dealing with nonpolygonal contact regions. We will assume that the motion constraints of the assembly can be expressed by a finite number of constraint inequalities in the form of equation (1), all of which must be satisfied. (Palmer[6] discusses the complexities that arise when only a subset of the motion constraints need to be satisfied by a given motion.)

For the remainder of the paper then, we will consider assemblies whose motion constraints are expressed in terms of m contact points between the bodies. Let the ith contact point of the assembly be a contact between bodies A and B at the point $\mathbf{d}_i$ in a global frame of reference. Let $\hat{\mathbf{n}}_i$ denote the unit surface normal, pointing outwards from B towards A at $\mathbf{d}_i$. Let $\mathbf{c}_a$ and $\mathbf{c}_b$ denote the position of the center of mass of bodies A and B respectively. If A undergoes a displacement $\delta\mathbf{p}_a = (\delta\mathbf{r}_a, \delta\boldsymbol{\theta}_a)$ then $\mathbf{d}_i$, as attached to A, undergoes the displacement

$$\delta\mathbf{r}_a + \delta\boldsymbol{\theta}_a \times (\mathbf{d}_i - \mathbf{c}_a).$$

Similarly, for a displacement $\delta\mathbf{p}_b = (\delta\mathbf{r}_b, \delta\boldsymbol{\theta}_b)$ of body B, $\mathbf{d}_i$'s displacement, as attached to B, is

$$\delta\mathbf{r}_b + \delta\boldsymbol{\theta}_b \times (\mathbf{d}_i - \mathbf{c}_b).$$

The motion constraint at the ith contact point is therefore

$$\hat{\mathbf{n}}_i \cdot (\delta\mathbf{r}_a + \delta\boldsymbol{\theta}_a \times (\mathbf{d}_i - \mathbf{c}_a) - \delta\mathbf{r}_b - \delta\boldsymbol{\theta}_b \times (\mathbf{d}_i - \mathbf{c}_b)) \geq 0. \tag{2}$$

Note that if contact occurs between a body A and a fixed object B, we write a constraint without referring to a displacement of the fixed body, as

$$\hat{\mathbf{n}}_i \cdot (\delta\mathbf{r}_a + \delta\boldsymbol{\theta}_a \times (\mathbf{d}_i - \mathbf{c}_a)) \geq 0. \tag{3}$$

To simplify bookkeeping, we do not count fixed objects as bodies in our assembly; rather, we simply note when regular movable objects are in contact with fixed objects, and generate the appropriate motion constraint, such as equation (3).

Since each constraint is a linear inequality on the $\delta\mathbf{r}$ and $\delta\boldsymbol{\theta}$ variables, we can express the simultaneous satisfaction of all the constraints as one large linear system. If the vector $\delta\mathbf{p}$ denotes the virtual displacements of an assembly with n bodies, that is

$$\delta\mathbf{p} = \begin{pmatrix} \delta\mathbf{r}_1 \\ \delta\boldsymbol{\theta}_1 \\ \vdots \\ \delta\mathbf{r}_n \\ \delta\boldsymbol{\theta}_n \end{pmatrix},$$

then we express all m motion constraints by writing

$$\mathbf{J}\delta\mathbf{p} \geq \mathbf{0} \tag{4}$$

where $\mathbf{J}$ is an $m \times 6n$ matrix (since each displacement pair $(\delta\mathbf{r}_j, \delta\boldsymbol{\theta}_j)$ has six components) and $\mathbf{0}$ is an appropriately sized vector of zeroes. (We will denote row vectors, column vectors, and matrices whose entries are all zero simply by $\mathbf{0}$ throughout this paper. The dimension of $\mathbf{0}$ should be clear from the context in which it occurs.) The coefficients of $\mathbf{J}$ are computed according to the constraint equations (2) and (3).

Using this notation, we can say that a legal motion for an assembly is a displacement $\delta\mathbf{p}$ that satisfies $\mathbf{J}\delta\mathbf{p} \geq \mathbf{0}$. Note that the displacement $\delta\mathbf{p} = \mathbf{0}$ always yields a legal motion (the null-motion). Also, if $\delta\mathbf{p} \neq \mathbf{0}$ is a legal motion and α is a nonnegative scalar, then $\alpha\delta\mathbf{p}$ is also a legal motion, but $-\alpha\delta\mathbf{p}$ is not.

3 Determining Stability

Determining if an assembly is stable under a particular direction of gravity is fairly straightforward. In this section, we describe two different methods for determining if an assembly is stable. The first method is based on potential energy considerations, while the second method considers the contact forces that arise at contact points. Both methods involve linear programming. The latter method was used in work by Blum, Griffith, and Neumann[3], and is not limited to frictionless assemblies. In the next section, we show how both of these methods for determining stability can be modified to find a stable orientation for the assembly (if it exists).

3.1 Potential Energy

Suppose that an assembly undergoes a virtual displacement $\delta\mathbf{p}$. The change in potential energy δU corresponding to the displacements $\delta\mathbf{p}_i = (\delta\mathbf{r}_i, \delta\boldsymbol{\theta}_i)$ of the n bodies is

$$\delta U = -\sum_{i=1}^{n} M_i \mathbf{g} \cdot \delta\mathbf{r}_i \tag{5}$$

where M_i is the mass of the ith body. If we define the matrix $\mathbf{M}$ as the $3 \times 6n$ matrix

$$\mathbf{M} = \begin{pmatrix} M_1 & 0 & 0 & 0 & 0 & 0 & \ldots & M_n & 0 & 0 & 0 & 0 & 0 \\ 0 & M_1 & 0 & 0 & 0 & 0 & \ldots & 0 & M_n & 0 & 0 & 0 & 0 \\ 0 & 0 & M_1 & 0 & 0 & 0 & \ldots & 0 & 0 & M_n & 0 & 0 & 0 \end{pmatrix}$$

then we can write

$$\sum_{i=1}^{n} M_i \delta\mathbf{r}_i = \mathbf{M}\delta\mathbf{p} \tag{6}$$

and thus

$$\delta U = -\mathbf{g} \cdot \mathbf{M}\delta\mathbf{p} = -\mathbf{g}^T\mathbf{M}\delta\mathbf{p}. \tag{7}$$

As stated in the introduction, if for a given gravity direction $\mathbf{g}$ all legal motions yield $\delta U \geq 0$, then the assembly is stable. We are therefore interested in knowing the minimum value that δU can assume over all legal motions. If we let $\bar{z}$ denote this minimum, by writing

$$\bar{z} = \min_{\mathbf{J}\delta\mathbf{p} \geq \mathbf{0}} -\mathbf{g}^T\mathbf{M}\delta\mathbf{p} \tag{8}$$

then the structure is stable if $\bar{z} \geq 0$. Note however that the null-motion $\delta\mathbf{p} = \mathbf{0}$ is always legal, and yields $\delta U = 0$. Thus, $\bar{z}$ is bounded above by zero, and we can say simply that the structure is stable if $\bar{z} = 0$.

The value $\bar{z}$ can be determined by linear programming. However, as it stands, if $\bar{z} \neq 0$, then there must exist a legal $\delta\mathbf{p}$ for which $\delta U < 0$. In this case, the minimum value of δU is $-\infty$, since a displacement of $\alpha\delta\mathbf{p}$ is legal and yields an energy change of $\alpha\delta U < 0$ for any $\alpha > 0$. This is a consequence of the constraint $\mathbf{J}\delta\mathbf{p} \geq \mathbf{0}$, which constrains the motion direction $\delta\mathbf{p}$, but not its magnitude.

Anticipating future development, it is useful to bound the magnitude of the displacements $\delta\mathbf{p}$ considered in equation (8). Since we would like to be able to use linear programming techniques, we would like to bound $\delta\mathbf{p}$'s magnitude with linear constraints. We can do this straightforwardly by redefining $\bar{z}$ as

$$\bar{z} = \min_{\substack{\mathbf{J}\delta\mathbf{p} \geq \mathbf{0} \\ \|\delta\mathbf{p}\|_\infty \leq 1}} -\mathbf{g}^T\mathbf{M}\delta\mathbf{p}. \tag{9}$$

The infinity norm $||\mathbf{v}||_\infty$ of a vector $\mathbf{v}$ is the maximum absolute value over all the components of $\mathbf{v}$. The condition $||\delta\mathbf{p}||_\infty \leq 1$ constrains all components of $\delta\mathbf{p}$ to have magnitude less than one. Equation (9) can then be solved by linear programming. If the solution $\bar{z}$ is zero, then the assembly is stable. Otherwise, $\bar{z}$ is a (finite) negative value, and the assembly is unstable. (Note however that the displacement $\delta\mathbf{p}$ which yields the minimal $\bar{z} = \delta U$ only approximately indicates the direction of impending motion of the assembly. In order to exactly determine the impending motion direction, it is necessary to solve a quadratic programming problem[1].)

3.2 Contact Forces

Instead of looking at motion directions which decrease potential energy, we can consider the contact force that arises at each of the m contact points of the assembly. Since we are dealing with frictionless contacts, we know that the contact forces will act normal to the contact surfaces. Thus, at the ith contact point, we consider a contact force $f_i\hat{\mathbf{n}}_i$ that acts on body A of the contact, and a contact force $-f_i\hat{\mathbf{n}}_i$ that acts on body B of the contact, with f_i the unknown scalar magnitude of the force. Since $\hat{\mathbf{n}}_i$ is directed from B towards A, and since contact forces must be repulsive, the magnitude f_i must be nonnegative; that is, $f_i \geq 0$.

Let the vector of contact force magnitudes f_i be denoted by $\boldsymbol{f}$. The net force $\mathbf{F}_j \in \mathbf{R}^3$ acting on the jth body of the assembly can be written as

$$\mathbf{F}_j = \sum_{i=1}^{n} s_{ji} f_i \hat{\mathbf{n}}_i + M_i \mathbf{g} \tag{10}$$

where s_{ji} is either 1, -1, or zero. If the jth body is not involved in the ith contact, then s_{ji} is zero. If the contact force exerted on the jth body from the ith contact point is $f_i\hat{\mathbf{n}}_i$, then s_{ji} is 1. Otherwise, the contact force acting on the jth body is $-f_i\hat{\mathbf{n}}_i$, and s_{ji} is -1.

The net torque $\boldsymbol{\tau}_j \in \mathbf{R}^3$ acting on the jth body of the assembly is similarly written as

$$\boldsymbol{\tau}_j = \sum_{i=1}^{n} s_{ji}(\mathbf{d}_i - \mathbf{c}_j) \times f_i \hat{\mathbf{n}}_i \tag{11}$$

where $\mathbf{d}_i$ is the location of the ith contact point, and $\mathbf{c}_j$ is the location of the center of mass of the jth body. The scalars s_{ji} are the same as in the previous equation. The $\boldsymbol{\tau}_j$ are independent of $\mathbf{g}$ since a uniform gravity field does not exert a torque. If we define the $6n$-vectors $\mathbf{Q}$ and $\mathbf{G}$ as the collections

$$\mathbf{Q} = \begin{pmatrix} \mathbf{F}_1 \\ \boldsymbol{\tau}_1 \\ \vdots \\ \mathbf{F}_n \\ \boldsymbol{\tau}_n \end{pmatrix} \quad \text{and} \quad \mathbf{G} = \begin{pmatrix} M_1\mathbf{g} \\ \mathbf{0} \\ \vdots \\ M_n\mathbf{g} \\ \mathbf{0} \end{pmatrix}, \tag{12}$$

we can write

$$\mathbf{Q} = \mathbf{A}\boldsymbol{f} + \mathbf{G} \tag{13}$$

where $\mathbf{A}$ is a $6n \times m$ matrix whose coefficients are given by equations (10) and (11).

Because the assembly is frictionless, its impending motion is completely determined[4]. If there exist repulsive contact forces such that the net force and torque on every body is zero, then such forces will arise at the contact points, and the assembly is stable and will not move. Thus, we can determine stability by simply checking whether there exists $\boldsymbol{f}$ such that

$$\mathbf{Q} = \mathbf{A}\boldsymbol{f} + \mathbf{G} = \mathbf{0} \quad \text{and} \quad \boldsymbol{f} \geq \mathbf{0}. \tag{14}$$

The existence of a suitable $\boldsymbol{f}$ can be determined by linear programming.

4 Finding a Stable Orientation

Suppose that for an assembly and a given gravity direction we find that the structure is unstable, using either of the two methods described in the previous section. Can we find a new direction for $\mathbf{g}$ which will make the structure stable? In this section, we show how the contact-force formulation to determine stability can be trivially modified to find a value for $\mathbf{g}$ (if it exists) which makes the assembly stable.

In section 3.2, we were searching for a vector $\boldsymbol{f}$ so that

$$\mathbf{Q} = \mathbf{A}\boldsymbol{f} + \mathbf{G} = \mathbf{0} \quad \text{and} \quad \boldsymbol{f} \geq \mathbf{0} \tag{15}$$

where $\mathbf{G}$ depended upon the known value of $\mathbf{g}$. If however we treat $\mathbf{g}$ as an additional unknown, all we need to do is simply check and see if there exist values for $\boldsymbol{f}$ *and* $\mathbf{g}$ that satisfy $\mathbf{Q} = \mathbf{0}$, indicating that the assembly is stable. That is, if we can find vectors $\mathbf{g}$ and $\boldsymbol{f}$ such that

$$\mathbf{Q} = \mathbf{A}\boldsymbol{f} + \mathbf{G} = \mathbf{0}, \quad \boldsymbol{f} \geq \mathbf{0} \quad \text{and} \quad \|\mathbf{g}\|_2 = 1 \tag{16}$$

(where $\mathbf{G}$ is defined in terms of $\mathbf{g}$ by equation (12)) then the assembly is stable in orientation $\mathbf{g}$. If equation (16) has a solution, we can find it by linear programming, although a slight modification is required.

The constraint that $\mathbf{g}$ be a unit vector (that is, $\|\mathbf{g}\|_2 = 1$) cannot be enforced in a linear program. However, it is not necessary to search among only unit vectors; it is merely necessary to make sure that we search among all possible directions. We can do this by considering vectors $\mathbf{g}$ such that $\|\mathbf{g}\|_1 = 1$. (For a vector $\mathbf{v}$, $\|\mathbf{v}\|_1 = \sum_i |v_i|$.) However, since the set of vectors $\|\mathbf{g}\|_1 = 1$ is nonconvex, we need to split it up into pieces. Intuitively, $\|\mathbf{g}\|_1 = 1$ forms a "unit diamond" about the origin, consisting of eight planar facets. Let the set S_1 be defined as the set of vectors $\mathbf{g} = (g_x, g_y, g_z)^T$ satisfying

$$g_x, g_y, g_z \geq 0 \quad \text{and} \quad g_x + g_y + g_z = 1. \tag{17}$$

Similarly, define S_2 to consist of vectors satisfying

$$-g_x, g_y, g_z \geq 0 \quad \text{and} \quad -g_x + g_y + g_z = 1 \tag{18}$$

and so on through all the eight sign permutations of g_x, g_y and g_z. Using this notation, we can look for a $\mathbf{g}$ that makes the assembly stable by seeing if any of the eight linear programs

$$\mathbf{Q} = \mathbf{A}\boldsymbol{f} + \mathbf{G} = \mathbf{0}, \quad \boldsymbol{f} \geq \mathbf{0}, \quad \text{and} \quad \mathbf{g} \in S_i \ \ (1 \leq i \leq 8) \tag{19}$$

have a solution f and $\mathbf{g}$.[1]

Note that if an assembly can be made stable, we find the value of $\mathbf{g}$ which makes it stable. However, when no such $\mathbf{g}$ exists, all we know is that the linear program is unsatisfiable. Even if an assembly has no stable orientation, it would still be desirable to know what orientation $\mathbf{g}$ comes the closest to making the assembly stable. In the next section, we will show how such a direction $\mathbf{g}$ can be found by linear programming using the potential energy formulation.

5 Finding the Most Stable Orientation

Modifying the contact-force formulation to find a stable orientation (if it exists) was straightforward. In working with the potential energy formulation though, we are not limited to simply finding a stable orientation, or reporting that the assembly is unstable. Instead, we can modify the potential energy formulation so that we can find either a stable orientation, or, for unstable assemblies, the most stable orientation possible.

Let us recast $\bar{z}$ as a function of $\mathbf{g}$ by writing

$$\bar{z}(\mathbf{g}) = \min_{\substack{\mathbf{J}\delta\mathbf{p} \geq \mathbf{0} \\ \|\delta\mathbf{p}\|_\infty \leq 1}} -\mathbf{g}^T \mathbf{M} \delta\mathbf{p}. \tag{20}$$

If $\mathbf{g}$ is an orientation for which the assembly is stable, then $\bar{z}(\mathbf{g}) = 0$. Otherwise, the assembly is unstable, and $\bar{z}(\mathbf{g}) < 0$. We will use the function $\bar{z}(\mathbf{g})$ as a measure of the instability of an assembly under gravity $\mathbf{g}$. Since there may be no value $\mathbf{g}$ for which $\bar{z}(\mathbf{g}) = 0$, we will search for a value of $\mathbf{g}$ that maximizes $\bar{z}(\mathbf{g})$. If this maximum is zero, then we will have found a stable orientation. Otherwise, we will have found the "most" stable orientation, as defined by the metric $\bar{z}(\mathbf{g})$. In the remainder of this paper, we will restrict $\mathbf{g}$ to lie in S_1, by writing $\mathbf{g} \geq \mathbf{0}$ and $\|\mathbf{g}\|_1 = \sum_{i=1}^3 g_i = 1$. In searching for the most stable assembly, we will have to perform eight different searches; one for each partition S_i. All statements and methods made hereafter involving $\mathbf{g} \in S_1$ can be applied to the other seven cases of $\mathbf{g} \in S_i$.

5.1 Maximin

To find the most stable orientation, we are trying to solve a *maximin* problem. That is, we are trying to solve

$$\max_{\substack{\mathbf{g} \geq \mathbf{0} \\ \sum g_i = 1}} \bar{z}(\mathbf{g}) = \max_{\substack{\mathbf{g} \geq \mathbf{0} \\ \sum g_i = 1}} \left(\min_{\substack{\mathbf{J}\delta\mathbf{p} \geq \mathbf{0} \\ \|\delta\mathbf{p}\|_\infty \leq 1}} -\mathbf{g}^T \mathbf{M} \delta\mathbf{p} \right). \tag{21}$$

[1] We could also partition gravity by considering the unit cube of directions $\|\mathbf{g}\|_\infty = 1$. The natural division here would be to use sets S_1 through S_6, with S_1 defined by $g_x = 1$ and $-1 \leq g_y, g_z \leq 1$ and S_2 by $g_x = -1$ and $-1 \leq g_y, g_z \leq 1$, and similarly for S_3 through S_6. This would work just as well as considering the set $\|\mathbf{g}\|_1 = 1$. The only reason for using the metric $\|\mathbf{g}\|_1 = 1$ in this section is that we are forced to use this metric in subsequent sections.

Constrained maximin problems are in general hard to solve. However, problem (21) has a form similar to a maximin problem that is solvable by linear programming. Given an $m \times n$ matrix $\mathbf{A}$, the minimax theorem of matrix games, first proved by Von Neumann, states that

$$\max_{\substack{\mathbf{x} \geq \mathbf{0} \\ \sum x_i = 1}} \left(\min_{\substack{\mathbf{y} \geq \mathbf{0} \\ \sum y_i = 1}} \mathbf{y}^T \mathbf{A} \mathbf{x} \right) = \min_{\substack{\mathbf{y} \geq \mathbf{0} \\ \sum y_i = 1}} \left(\max_{\substack{\mathbf{x} \geq \mathbf{0} \\ \sum x_i = 1}} \mathbf{y}^T \mathbf{A} \mathbf{x} \right). \tag{22}$$

where $\mathbf{x}$ and $\mathbf{y}$ are vectors in $\mathbf{R}^m$ and $\mathbf{R}^n$ respectively. Furthermore, the value of $\mathbf{x}$ for which the maximum on the left is attained can be found by solving a linear program; the solution of the dual linear program gives the value of $\mathbf{y}$ for which the minimum on the right is attained.

The maximin problem problem (21) is similar to problem (22) in that the constraints on $\mathbf{g}$ are $\mathbf{g} \geq \mathbf{0}$ and $\sum g_i = 1$. However, the constraints on the inner variables $\delta\mathbf{p}$ have quite a different form. It turns out that a variant of the linear program used to solve problem (22) for $\mathbf{x}$ can be used to solve problem (21) for $\mathbf{g}$. In section 6 we will exhibit a pair of dual linear programs which find a solution $\mathbf{g}$ to problem (21) and a solution $\delta\mathbf{p}$ to the dual problem

$$\min_{\substack{\mathbf{J}\delta\mathbf{p} \geq \mathbf{0} \\ \|\delta\mathbf{p}\|_\infty \leq 1}} \left(\max_{\substack{\mathbf{g} \geq \mathbf{0} \\ \sum g_i = 1}} -\mathbf{g}^T \mathbf{M} \delta\mathbf{p} \right). \tag{23}$$

It will also turn out that

$$\max_{\substack{\mathbf{g} \geq \mathbf{0} \\ \sum g_i = 1}} \bar{z}(\mathbf{g}) = \max_{\substack{\mathbf{g} \geq \mathbf{0} \\ \sum g_i = 1}} \left(\min_{\substack{\mathbf{J}\delta\mathbf{p} \geq \mathbf{0} \\ \|\delta\mathbf{p}\|_\infty \leq 1}} -\mathbf{g}^T \mathbf{M} \delta\mathbf{p} \right) = \min_{\substack{\mathbf{J}\delta\mathbf{p} \geq \mathbf{0} \\ \|\delta\mathbf{p}\|_\infty \leq 1}} \left(\max_{\substack{\mathbf{g} \geq \mathbf{0} \\ \sum g_i = 1}} -\mathbf{g}^T \mathbf{M} \delta\mathbf{p} \right). \tag{24}$$

For now however, it will be more instructive to simply hope that equation (24) holds, and use physical intuition to formulate a linear program that solves equation (23) for $\delta\mathbf{p}$. Assuming that equation (24) holds, the solution of the dual of this linear program will yield a vector $\mathbf{g}$ which maximizes problem (21). The intuition which allows us to directly formulate the necessary linear program lies in viewing problem (23) as a competition, or game (just as problem (22) is viewed as what is known as a "two-person zero-sum game").

5.2 A Particle versus Gravity

For illustrative purposes, let us greatly simplify the problem. Our assembly, for the moment, consists of a single particle in $\mathbf{R}^2$, with degrees of freedom δr_x and δr_y and unit mass. Our gravity vector likewise has two components g_x and g_y. To begin with, we will assume that there are no constraints on δr_x or δr_y, except for the bounds $|\delta r_x| \leq 1$ and $|\delta r_y| \leq 1$. We will search for the most stable orientation by finding the solution of

$$\min_{|\delta r_x|, |\delta r_y| \leq 1} \left(\max_{\substack{g_x, g_y \geq 0 \\ g_x + g_y = 1}} -L \right)$$

where

$$L = \begin{pmatrix} g_x \\ g_y \end{pmatrix} \cdot \begin{pmatrix} \delta r_x \\ \delta r_y \end{pmatrix} = g_x \delta r_x + g_y \delta r_y.$$

Let us simplify this by removing the minus sign; this swaps the "min" and "max" functions, yielding

$$\max_{|\delta r_x|, |\delta r_y| \leq 1} \left(\min_{\substack{g_x, g_y \geq 0 \\ g_x + g_y = 1}} L \right) \tag{25}$$

Normally, we think of forces such as gravity as having an "upsetting" effect on objects in that they cause objects to move unless balanced by other forces. Objects, on the other hand, have inertia—they seek to stay where they are unless compelled to move by forces. We can view equation (25) as a game by changing our viewpoint slightly. We will regard both the force of gravity, and the particle, as players in a game, each with a particular purpose. On one side, we have player "Particle." Rather than stand still, Player Particle would like to move as much as possible in the direction of gravity chosen by Player Gravity. Mathematically, player Particle seeks to maximize L. As a result, Particle is very lazy. While Particle is quite happy to make L positive by moving downhill (that is, in the direction chosen by Gravity), Particle is so lazy that rather than move uphill the slightest bit, making L negative, Particle would rather just stand still.

Conversely, Player Gravity's goal is to pick a gravity direction so that L is as small as possible. In terms of the game, Gravity needs to pick a nonzero direction that causes the *particle to want to move* as little as possible (which, as we said, is opposite our normal depiction of gravity as an upsetting force on systems). Gravity is therefore annoyed when the particle actually moves in the direction of gravity, since this makes L negative. Gravity would prefer the particle not to move at all, which makes L zero.

Now suppose that the game is played without any constraints on Particle's motion. (Clearly, Particle has the advantage here.) Particle reasons that by not moving at all (that is, by choosing $\delta r_x = \delta r_y = 0$), a value of $L = 0$ is attained. However, since Particle knows that Gravity cannot choose negative values for g_x or g_y, Particle chooses $\delta r_x = \delta r_y = 1$. (If Gravity wasn't restricted in this way, it would be quite a different story, and Particle would be advised to stick with $\delta r_x = \delta r_y = 0$.) As it is though, if Particle picks $\delta r_x = \delta r_y = 1$ then no matter what Gravity chooses, $L = 1$ and Particle wins, getting to move downhill. (As we said, this was a pretty lopsided game.)

Suppose however that Particle is in contact with an obstacle and must satisfy the motion constraint

$$\delta r_x + \delta r_y \leq 0 \tag{26}$$

along with the regular bounds $|\delta r_x| \leq 1$ and $|\delta r_y| \leq 1$. What does Gravity do in this case?

Gravity reasons as follows: "No matter what direction I pick, Particle always wants to go in that direction. Well, I'll fool Particle this time—I'll just pick a direction Particle *can't* go in. I'll choose $g_x = g_y = 1/2$. That way, I can guarantee that $L \leq 0$—because if Particle is foolish enough to make $\delta r_x > 0$ and thus $\delta r_y < 0$, I'll switch to $g_y = 1$. And likewise, if Particle makes $\delta r_x < 0$, I'll make $g_x = 1$. But wait! What if after I pick $g_x = 1$, Particle picks a new direction so that $L > 0$? Well, I can at least guarantee that Particle

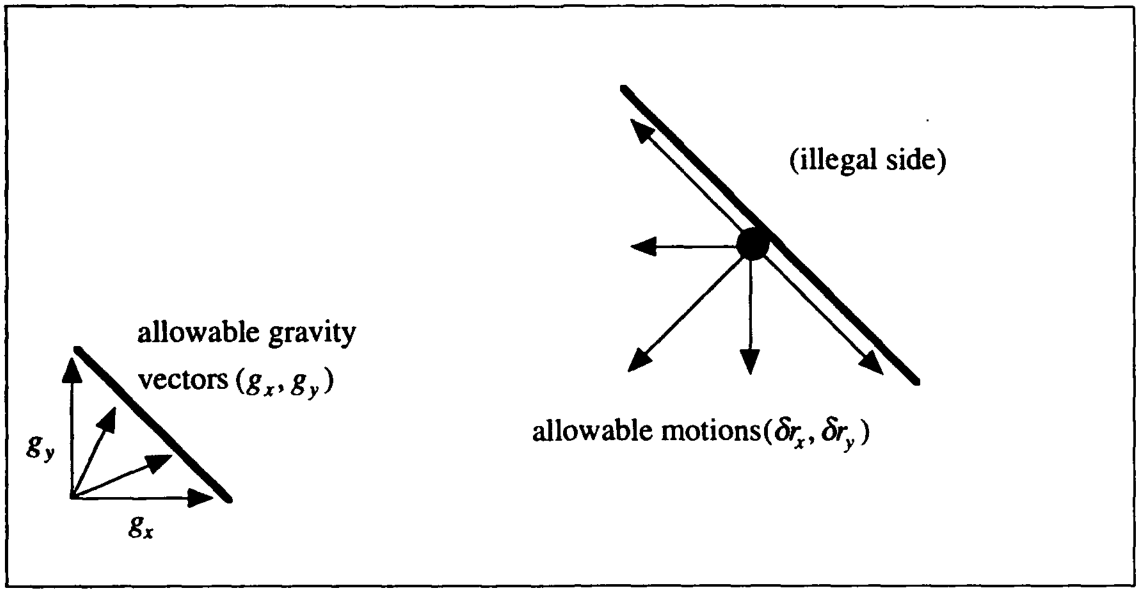

Figure 4: The motion of the particle is constrained because it is in contact with a wall. The optimum strategies are to choose $g_x = g_y = 1/2$, and $\delta r_x = \delta r_y = 0$.

can't move downhill by sticking with $g_x = g_y = 1/2$. There's *no* way Particle can move in that direction at all."

For its part, Particle reasons: "If I choose $\delta r_x < \delta r_y$, then Gravity will pick $g_x = 1$, to make L as small as possible. But if I choose $\delta r_y < \delta r_x$, then Gravity will pick $g_y = 1$. Aha! I'd better make it so that the smaller of the δr_x and δr_y is as *big* as possible. But wait! If I choose $\delta r_x > 0$, then I have to pick $\delta r_y < 0$ because of my motion constraint. And if I choose $\delta r_y > 0$, then I'd have to make $\delta r_x < 0$. Either way then, I'd be moving uphill, which means L would be negative. Forget it! I'll just choose $\delta r_x = \delta r_y = 0$ and not move at all!"

Particle's strategy is particularly easy to formulate as a linear program. Knowing that Gravity will focus attention on the smaller of δr_x or δr_y (and thus achieving a value of L equal to the smaller of the two) Particle's strategy is simply to choose δr_x and δr_y such that the *smaller* of the two is as *large* as possible (given the constraint $\delta r_x + \delta r_y \leq 0$).

Now let us apply this reasoning to equation (24), but with the minus sign removed:

$$\max_{\substack{\mathbf{J}\delta\mathbf{p} \geq 0 \\ \|\delta\mathbf{p}\|_\infty \leq 1}} \left(\min_{\substack{\mathbf{g} \geq 0 \\ \sum g_i = 1}} \mathbf{g}^T \mathbf{M} \delta\mathbf{p} \right). \tag{27}$$

Consider the vector $\mathbf{M}\delta\mathbf{p}$, which has length three (assuming we are working with a three-dimensional assembly). Given a vector $\delta\mathbf{p}$, the $\mathbf{g}$ that minimizes $\mathbf{g}^T\mathbf{M}\delta\mathbf{p}$ will be such that $g_i = 1$ where $(\mathbf{M}\delta\mathbf{p})_i$ is the smallest element of the vector $\mathbf{M}\delta\mathbf{p}$. Clearly then, the maximum of equation (27) occurs when $\delta\mathbf{p}$ is chosen so that the *minimum* component of $\mathbf{M}\delta\mathbf{p}$ is *maximized*. Such a $\delta\mathbf{p}$ can be found by linear programming. Then, assuming equation (24) is true, the solution to the dual of this linear program will yield the choice of $\mathbf{g}$ which maximizes $\bar{z}(\mathbf{g})$. In the next section, we explicitly describe a dual pair of linear programs that enable us to maximize $\bar{z}(\mathbf{g})$ and prove that equation (24) holds.

6 Linear Programming Solutions of the Maximin Problem

In this section, we exhibit dual linear programs to solve equations (21) and (23), but with the minus sign removed. That is, we will find a solution vector $\mathbf{g}$ to the problem

$$\min_{\substack{\mathbf{g}\geq\mathbf{0}\\ \Sigma g_i=1}} \left(\max_{\substack{\mathbf{J}\delta\mathbf{p}\geq\mathbf{0}\\ \|\delta\mathbf{p}\|_\infty\leq 1}} L \right) \tag{28}$$

and a solution vector $\delta\mathbf{p}$ to the problem

$$\max_{\substack{\mathbf{J}\delta\mathbf{p}\geq\mathbf{0}\\ \|\delta\mathbf{p}\|_\infty\leq 1}} \left(\min_{\substack{\mathbf{g}\geq\mathbf{0}\\ \Sigma g_i=1}} L \right) \tag{29}$$

where $L = \mathbf{g}^T\mathbf{M}\delta\mathbf{p}$. The goal is to make the optimal solutions to the dual linear programs satisfy a condition called a "saddle-point condition." A pair of vectors $\mathbf{g}^*$ and $\delta\mathbf{p}^*$ satisfying the constraints on $\mathbf{g}$ and $\delta\mathbf{p}$ in problems (28) and (29) is called a saddle-point if for all $\mathbf{g}$ and $\delta\mathbf{p}$ which also satisfy the constraints of the problems, the relation

$$\mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p} \leq \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}^* \leq \mathbf{g}^T\mathbf{M}\delta\mathbf{p}^*$$

holds. In terms of the game of the previous section, a saddle-point indicates a pair of strategies such that neither player is inclined to change their strategy, providing the other player holds constant as well. We will show that the solutions to the dual linear programs satisfy the saddle-point condition, and then prove that any vectors that satisfy the saddle-point condition solve problems (28) and (29).

Let us define the vector $\mathbf{b}$ by $\mathbf{b} = (1,1,1)^T$. In what follows, a *feasible* vector $\mathbf{g}$ is a vector satisfying $\mathbf{g} \geq \mathbf{0}$ and $\sum g_i = \mathbf{b}^T\mathbf{g} = 1$. To express feasibility for a vector $\delta\mathbf{p}$, let $\mathbf{I}$ denote the $6n \times 6n$ identity matrix, and let $\mathbf{e}$ be a vector of length $6n$, with every element equal to one. If we define the vector $\mathbf{d}$ of length $m + 12n$ and the $(m + 12n) \times 6n$ matrix $\mathbf{B}$ by

$$\mathbf{d} = \begin{pmatrix} \mathbf{0} \\ \mathbf{e} \\ \mathbf{e} \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} -\mathbf{J} \\ \mathbf{I} \\ -\mathbf{I} \end{pmatrix} \tag{30}$$

then $\mathbf{B}\delta\mathbf{p} \leq \mathbf{d}$ implies

$$\mathbf{B}\delta\mathbf{p} = \begin{pmatrix} -\mathbf{J}\delta\mathbf{p} \\ \delta\mathbf{p} \\ -\delta\mathbf{p} \end{pmatrix} \leq \begin{pmatrix} \mathbf{0} \\ \mathbf{e} \\ \mathbf{e} \end{pmatrix}. \tag{31}$$

This in turn implies $\mathbf{J}\delta\mathbf{p} \geq \mathbf{0}$ and $-\mathbf{e} \leq \delta\mathbf{p} \leq \mathbf{e}$ which is equivalent to $\|\delta\mathbf{p}\|_\infty \leq 1$. Thus, we will say that a vector $\delta\mathbf{p}$ is feasible if $\mathbf{B}\delta\mathbf{p} \leq \mathbf{d}$.

6.1 Primal and Dual Linear Programs

Let ν be a scalar and consider the (primal) linear program

$$\max \nu \qquad \text{subject to} \qquad \begin{pmatrix} \mathbf{b} & -\mathbf{M} \\ \mathbf{0} & \mathbf{B} \end{pmatrix} \begin{pmatrix} \nu \\ \delta\mathbf{p} \end{pmatrix} \leq \begin{pmatrix} \mathbf{0} \\ \mathbf{d} \end{pmatrix}. \tag{32}$$

If we view the constraint $\mathbf{b}\nu - \mathbf{M}\delta\mathbf{p} \leq 0$ as $\mathbf{M}\delta\mathbf{p} \geq \mathbf{b}\nu$, we see that to maximize ν we need to make $\mathbf{M}\delta\mathbf{p}$ as large as possible. In particular, ν is bounded by the maximum that the smallest element of $\mathbf{M}\delta\mathbf{p}$ can attain, given the constraints on $\delta\mathbf{p}$. Thus, this linear program exactly captures the strategy articulated at the end of section 5.2 to choose $\delta\mathbf{p}$. Since setting $\nu = 0$ and $\delta\mathbf{p} = \mathbf{0}$ satisfies the conditions in problem (32), it is clear that a solution to problem (32) exists. Note that for any pair $(\nu, \delta\mathbf{p})$ which satisfies the conditions in problem (32), $\mathbf{B}\delta\mathbf{p} \leq \mathbf{d}$ which implies that $\delta\mathbf{p}$ is feasible, as defined above.

The dual linear program to problem (32) is

$$\min\ (\mathbf{0}, \mathbf{d}^T) \begin{pmatrix} \mathbf{g} \\ \mathbf{s} \end{pmatrix} \qquad \text{subject to} \qquad (\mathbf{g}^T, \mathbf{s}^T) \begin{pmatrix} \mathbf{b} & -\mathbf{M} \\ \mathbf{0} & \mathbf{B} \end{pmatrix} = (1, \mathbf{0}),\ \mathbf{g}, \mathbf{s} \geq \mathbf{0} \tag{33}$$

where $\mathbf{s} \in \mathbf{R}^{m+12n}$. Note that any $\mathbf{g}$ satisfying the conditions in problem (33) satisfies $\mathbf{g} \geq \mathbf{0}$ and $\mathbf{g}^T\mathbf{b} = 1$, and is thus feasible.

6.2 Saddle-Point Condition

Let $\mathbf{g}^*$ and $\mathbf{s}^*$ be optimal solutions to problem (33) and let ν^* and $\delta\mathbf{p}^*$ be optimal solutions to problem (32). The duality theory of linear programming[7] states that the optimal values of the two linear programs are equal: that is,

$$\nu^* = (\mathbf{0}, \mathbf{d}^T) \begin{pmatrix} \mathbf{g}^* \\ \mathbf{s}^* \end{pmatrix} = \mathbf{d}^T\mathbf{s}^*. \tag{34}$$

The duality theory also states that

$$(\mathbf{g}^{*T}, \mathbf{s}^{*T}) \begin{pmatrix} \mathbf{M}\delta\mathbf{p}^* - \mathbf{b}\nu^* \\ \mathbf{d} - \mathbf{B}\delta\mathbf{p}^* \end{pmatrix} = 0. \tag{35}$$

The fact that both vectors in this equation are nonnegative (due to the conditions in the linear programs) means that

$$\mathbf{g}^{*T}(\mathbf{M}\delta\mathbf{p}^* - \mathbf{b}\nu^*) = 0 \tag{36}$$

or

$$\mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}^* = \mathbf{g}^{*T}\mathbf{b}\nu^* = \nu^* \tag{37}$$

since $\mathbf{g}^{*T}\mathbf{b} = 1$.

Since $\mathbf{g}^*$ and $\mathbf{s}^*$ satisfy the conditions in problem (33), $-\mathbf{g}^{*T}\mathbf{M} + \mathbf{s}^{*T}\mathbf{B} = \mathbf{0}$ or equivalently, $\mathbf{g}^{*T}\mathbf{M} = \mathbf{s}^{*T}\mathbf{B}$. For any feasible $\delta\mathbf{p}$ then,

$$(\mathbf{g}^{*T}\mathbf{M})\delta\mathbf{p} = (\mathbf{s}^{*T}\mathbf{B})\delta\mathbf{p} = \mathbf{s}^{*T}(\mathbf{B}\delta\mathbf{p}) \leq \mathbf{s}^{*T}\mathbf{d} \tag{38}$$

since $\mathbf{B}\delta\mathbf{p} \leq \mathbf{d}$ and $\mathbf{s}^*$ is nonnegative. Using the fact that $\mathbf{s}^{*T}\mathbf{d} = \nu^* = \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}^*$, we obtain

$$\mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p} \leq \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}^* \tag{39}$$

for all feasible $\delta\mathbf{p}$.

To obtain the other half of the saddle-point condition, observe that $\mathbf{M}\delta\mathbf{p}^* \geq \mathbf{b}\nu^*$. Since $\mathbf{g} \geq \mathbf{0}$ and $\mathbf{g}^T\mathbf{b} = 1$ for any feasible $\mathbf{g}$,

$$\mathbf{g}^T(\mathbf{M}\delta\mathbf{p}^*) \geq \mathbf{g}^T(\mathbf{b}\nu^*) = (\mathbf{g}^T\mathbf{b})\nu^* = \nu^* = \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}^*. \tag{40}$$

Combining the previous two inequalities gives the saddle-point condition

$$\mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p} \leq \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}^* \leq \mathbf{g}^T\mathbf{M}\delta\mathbf{p}^*. \tag{41}$$

for any feasible $\mathbf{g}$ and $\delta\mathbf{p}$.

6.3 Maximin Result

We claim that problem (28) attains its minimum value (and thus $\bar{z}(\mathbf{g})$ attains its maximum value) for $\mathbf{g} = \mathbf{g}^*$, and that problem (29) attains its maximum value for $\delta\mathbf{p} = \delta\mathbf{p}^*$. Furthermore, these values are equal. In what follows, we restrict our attention to feasible vectors $\delta\mathbf{p}$ and $\mathbf{g}$ (that is, we will not write out the feasibility conditions).

To show that $\mathbf{g}^*$ minimizes problem (28) we need to show that

$$\max_{\delta\mathbf{p}} \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p} \leq \max_{\delta\mathbf{p}} \mathbf{g}^T\mathbf{M}\delta\mathbf{p}. \tag{42}$$

for all feasible $\mathbf{g}$. From the saddle-point condition, $\mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p} \leq \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}^*$ for all feasible $\delta\mathbf{p}$, so $\max\limits_{\delta\mathbf{p}} \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}$ is bounded from above by $\mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}^*$. Since $\mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}^* \leq \mathbf{g}^T\mathbf{M}\delta\mathbf{p}^*$ for any feasible $\mathbf{g}$, we have

$$\max_{\delta\mathbf{p}} \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p} \leq \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}^* \leq \mathbf{g}^T\mathbf{M}\delta\mathbf{p}^* \leq \max_{\delta\mathbf{p}} \mathbf{g}^T\mathbf{M}\delta\mathbf{p} \tag{43}$$

for any feasible $\mathbf{g}$.[2] Thus,

$$\min_{\mathbf{g}}\left(\max_{\delta\mathbf{p}} \mathbf{g}^T\mathbf{M}\delta\mathbf{p}\right) = \max_{\delta\mathbf{p}} \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p} \leq \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}^*. \tag{44}$$

Similarly, the saddle-point condition yields

$$\min_{\mathbf{g}} \mathbf{g}^T\mathbf{M}\delta\mathbf{p} \leq \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p} \leq \mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}^* \leq \min_{\mathbf{g}} \mathbf{g}^T\mathbf{M}\delta\mathbf{p}^* \tag{45}$$

for all feasible $\delta\mathbf{p}$, so $\delta\mathbf{p}^*$ maximizes problem (29) and

$$\mathbf{g}^{*T}\mathbf{M}\delta\mathbf{p}^* \leq \min_{\mathbf{g}} \mathbf{g}^T\mathbf{M}\delta\mathbf{p}^* = \max_{\delta\mathbf{p}}\left(\min_{\mathbf{g}} \mathbf{g}^T\mathbf{M}\delta\mathbf{p}\right). \tag{46}$$

[2] The inequality $\mathbf{g}^T\mathbf{M}\delta\mathbf{p}^* \leq \max_{\delta\mathbf{p}} \mathbf{g}^T\mathbf{M}\delta\mathbf{p}$ follows from the fact that $\max_{\delta\mathbf{p}} \mathbf{g}^T\mathbf{M}\delta\mathbf{p}$ is greater than or equal to $\mathbf{g}^T\mathbf{M}\delta\mathbf{p}$ for *any* choice of $\delta\mathbf{p}$. In particular then if we choose $\delta\mathbf{p} = \delta\mathbf{p}^*$, $\max_{\delta\mathbf{p}} \mathbf{g}^T\mathbf{M}\delta\mathbf{p} \geq \mathbf{g}^T\mathbf{M}\delta\mathbf{p}^*$.

This proves that $\mathbf{g} = \mathbf{g}^*$ and $\delta\mathbf{p} = \delta\mathbf{p}^*$ are optimal for problems (28) and (29) respectively, and

$$\min_{\mathbf{g}} \left(\max_{\delta\mathbf{p}} \mathbf{g}^T \mathbf{M} \delta\mathbf{p} \right) \leq \mathbf{g}^{*T} \mathbf{M} \delta\mathbf{p}^* \leq \max_{\delta\mathbf{p}} \left(\min_{\mathbf{g}} \mathbf{g}^T \mathbf{M} \delta\mathbf{p} \right). \tag{47}$$

Thus, we can find a value of $\mathbf{g}$ which maximizes $\bar{z}(\mathbf{g})$ by solving the linear program of equation (33). The motion $\delta\mathbf{p} = \delta\mathbf{p}^*$ which minimizes δU for the orientation $\mathbf{g}^*$ according to the constraints $||\delta\mathbf{p}||_\infty \leq 1$ and $\mathbf{J}\delta\mathbf{p} \geq \mathbf{0}$ can be found by solving the linear program (32).

Finally, we can use the optimal vectors $\mathbf{g}^*$ and $\delta\mathbf{p}^*$ to obtain

$$\max_{\delta\mathbf{p}} \left(\min_{\mathbf{g}} \mathbf{g}^T \mathbf{M} \delta\mathbf{p} \right) = \min_{\mathbf{g}} \mathbf{g}^T \mathbf{M} \delta\mathbf{p}^* \leq \mathbf{g}^{*T} \mathbf{M} \delta\mathbf{p}^* \leq \max_{\delta\mathbf{p}} \mathbf{g}^{*T} \mathbf{M} \delta\mathbf{p} = \min_{\mathbf{g}} \left(\max_{\delta\mathbf{p}} \mathbf{g}^T \mathbf{M} \delta\mathbf{p} \right)$$

which yields

$$\max_{\delta\mathbf{p}} \left(\min_{\mathbf{g}} L \right) \leq \min_{\mathbf{g}} \left(\max_{\delta\mathbf{p}} L \right). \tag{48}$$

Combining the inequalities in (47) and(48) proves that

$$\min_{\mathbf{g}} \left(\max_{\delta\mathbf{p}} \mathbf{g}^T \mathbf{M} \delta\mathbf{p} \right) = \mathbf{g}^{*T} \mathbf{M} \delta\mathbf{p}^* = \max_{\delta\mathbf{p}} \left(\min_{\mathbf{g}} \mathbf{g}^T \mathbf{M} \delta\mathbf{p} \right). \tag{49}$$

References

[1] D. Baraff. Issues in computing contact forces for non-penetrating rigid bodies. *Algorithmica*, (to appear).

[2] D. Baraff and A. Witkin. Dynamic simulation of non-penetrating flexible bodies. In *Computer Graphics* (*Proc. SIGGRAPH*), volume 26, pages 303–308. ACM, July 1992.

[3] M. Blum, A. Griffith, and B. Neumann. A stability test for configurations of blocks. Technical Report TR AI Memo 188, Massachusetts Institute of Technology, 1970.

[4] W. Ingleton. A problem in linear inequalities. *Proceedings of the London Mathematical Society*, 16(3):519–536, 1966.

[5] R. Mattikalli, P. Khosla, B. Repetto, and D. Baraff. Stability of assemblies. In *International Conference on Intelligent Robots and Systems*, page (to appear). IEEE/RSJ, July 1993.

[6] R.S. Palmer. *Computational Complexity of Motion and Stability of Polygons*. PhD thesis, Cornell University, January 1987.

[7] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, 1986.